

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: LINKED CODE GENERATION REPORTS

APPLICANTS: KAI TUSCHNER and WILLIAM J. ALDRICH

Kenneth F. Kozik
Reg. No. 36,572
Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Tel.: (617) 542-5070
Fax: (617) 542-8906

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL 298 425 957 US

I hereby certify under 37 CFR \$1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

10-25-01
Date of Deposit

Heroy Jenkins
Signature

Heroy Jenkins
Typed or Printed Name of Person Signing Certificate

LINKED CODE GENERATION REPORTS

TECHNICAL FIELD

This invention relates to linked code generation reports.

5

BACKGROUND

10 Data representation and modeling are an integral part of
working with dynamic real-world systems such as electrical
circuits, shock absorbers, braking systems, and many other
electrical, mechanical and thermodynamic systems. These
15 systems may be modeled, simulated and analyzed on a computer
system using block diagram modeling. Block diagram modeling
graphically depicts mathematical relationships among a
system's inputs, states, parameters, and outputs, typically
through the use of a graphical user interface (GUI). Block
20 diagram modeling also graphically depicts time-dependent
mathematical relationships among a system's inputs, states and
outputs, typically for display on the GUI.

Block diagram modeling may involve automatic code
25 generation, a process whereby software source code is
automatically produced from a block diagram model of a dynamic
system. The software source code produced by the automatic
code generation process may be compiled and executed on a

target processor, implementing the functionality specified by the model.

It is common for the code generated by an automatic code generator to include explanatory comments. Typically, such comments act as headers to generated functions or data structures, and provide an indication of how those functions or data structures logically relate to the block diagram model.

SUMMARY

According to one aspect of the invention, a method includes generating source code corresponding to a block diagram model, and generating hypertext links associating elements of the generated source code with elements of the block diagram model.

One or more of the following features may also be included. The method may further include displaying the source code and hypertext links on a display, receiving input from a user representing the selection of one of the hypertext links, and displaying to the user at least a portion of the block diagram model including an element of the model associated with the hypertext link. Displaying may include displaying the associated element in a highlighted fashion. At least one of the associated elements in the generated

source code may be a commented reference to a block in the block diagram model. At least one of the associated elements in the generated source code may be a variable reference in an operative code section. The hypertext link may be Standard
5 Generalized Markup Language (SGML) or Hypertext Markup Language (HTML). The hypertext language may be Extensible Markup Language (XML). The commented reference to a block may include a character string identifying a path to a file providing information relating to the sections of the block.

10 According to another aspect of the invention, a system includes means for generating source code corresponding to a block diagram model and means for generating hypertext links associating elements of the generated source code with elements of the block diagram model.

15 One or more of the following features may also be included. The system may further include means for displaying the source code and hypertext links on a display, means for receiving input from a user representing the selection of one of the hypertext links, and means for displaying to the user
20 at least a portion of the block diagram model including an element of the model associated with the hypertext link. The means for displaying to the user at least a portion of the block diagram model may include displaying the associated element in a highlighted fashion. At least one of the

associated elements in the generated source code may be a commented reference to a block in the block diagram model. At least one of the associated elements in the generated source code may be a variable reference in an operative code section.

5 The hypertext link may be Standard Generalized Markup Language (SGML) or Hypertext Markup Language (HTML). The hypertext language may be Extensible Markup Language (XML). The commented reference to a block may include a character string identifying a path to a file providing information relating to the sections of the block.

10 Embodiments may have one or more of the following advantages.

15 The system provides easy access and direct navigation from a syntax-highlighted code generation report to a block in a block diagram model. The use of the syntax-highlighted code generation report enables the user to directly view and access blocks in a block diagram model from their corresponding source code within a browser's window while at the same time allowing the user to modify, reconfigure, and view other
20 relevant data pertaining to the blocks under analysis as the block is being viewed. Moreover, in addition to the ease of instant navigation, the traceability of the source code back to the corresponding block increases efficiency in data modeling, processing, and analysis, and is beneficial in

increasing accuracy of results and minimizing analytical human error.

The syntax-highlighted code generation report provides increased efficiency and operation in terms of the usage, storage, and management of information. Information about a particular dynamic system is quickly and efficiently accessed as well as managed and stored in an efficient manner.

The syntax-highlighted code generation report enables a user to efficiently control the information related to the data to be used with a block-based design environment, as opposed to having to exit, enter, re-enter, close or open new viewing windows to access source code information about the block. This facilitates a seamless transfer and view of information from the source code to the block-based design environment, enabling users to have a location with all the information required for analyzing and studying a block diagram model and/or system. Moreover, this reduces the redundancy, inconsistency and errors associated with having multiple sources of access to the source code. In addition, the ability to easily and quickly link the block to the source code makes it possible for users to reliably work without interruption, reducing inconsistency and redundancy.

Syntax-highlighted code generation reports simplify the task of defining, viewing, and modifying data associated with the block.

The source code accessed from the syntax-highlighted code generation report is available throughout the block-based design environment and is propagated to all stages of the process.

Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of a code generation report generating system.

FIG. 2 is a flow diagram of the link code generation report process of FIG. 1.

FIG. 3 is a diagram of the report compiler of FIG. 1.

FIG. 4 is a flow diagram of the parsing process of FIG. 3.

FIG. 5 illustrates an exemplary model diagram and associated syntax highlighted code generation report.

DETAILED DESCRIPTION

FIG. 1 shows a processing system 10. The processing system 10 includes a computer 12, such as a personal computer (PC). Computer 12 is connected to a network 14, such as the Internet, that runs TCP/IP (Transmission Control Protocol/Internet Protocol) or another suitable protocol. Connections may be via Ethernet, wireless link, telephone line, and the like.

Computer 12 contains a processor 16 and a memory 18. Memory 18 stores an operating system ("OS") 20 such as Windows98® or Linux, a TCP/IP protocol stack 22 for communicating over network 14, and machine-executable instructions 24 executed by processor 16 to perform linked code generation report process 100 below. The memory 18 also includes a code compiler 26 and a report compiler 28. Computer 12 also includes an input/output (I/O) device 30 for display of a graphical user interface (GUI) 32 to a user 34.

Referring to FIG. 2, the linked code generation report process 100 includes generating (102) a model diagram. The model diagram represents a dynamic system to be simulated and is displayed to the user 34 on the GUI 32 of the input/output device 30.

The model diagram is specified by the user 34 and represented by a source model language such as, for example,

Simulink® from The Mathworks, Inc. of Natick, MA, incorporated herein by reference. The process 100 converts (104) in the code compiler 26 the source model language into program source code in a technique generally referred to as code generation.

5 Code generation is a technique whereby software, i.e., program source code such as C, Ada, Basic and Java®, is automatically produced from the source model language representing by the model diagram. The software source code produced may be compiled and then executed on a target processor, implementing the functionality of the specified model diagram.

10 An example automatic code generator is Real Time Workshop® Embedded Coder from The Mathworks, Inc. of Natick, MA, incorporated herein by reference.

15 The process 100 generates (106) using the report compiler 28 a markup language document, generally referred to as a code generation report, that contains information about the source model language, settings of the code generator and the generated program source code in syntax highlighted form. Each part of the generated program source code is translated by the report compiler 28 and saved into its own markup language file. The markup language file is generally referred to as a syntax-highlighted code generation report.

20 The generated markup language files contain hyperlinks to the source model language representing the model diagram and

allow the user 34 to navigate from the markup language file to the source model language and the block it represents in the model diagram. This provides the user 34 with an ability to identify a block that corresponds to selected code fragments in the generated program source code.

A hyperlink is a selectable connection from one word, picture, or information object to another. In a multimedia environment such as the World Wide Web, such objects can include sound and motion video sequences. The most common form of link is the highlighted word or picture that can be selected by the user (with a mouse or in some other fashion), resulting in the immediate delivery and view of another file. The highlighted object is referred to as an anchor. The anchor reference and the object referred to constitute a hyperlink.

Referring to FIG. 3, the report compiler 28 includes a parsing process 110 and a software source code markup language conversion process 112. The parsing process 110 analyzes the generated software source code and replaces listed block references in the comment sections with links that refer back to the corresponding sections within the source model language representing the blocks of the model diagram. The software source code to markup language conversion process 112 converts the generated software source code to the syntax highlighted markup language code generation report.

The parsing process 110 is best understood by using a C source code example. Referring to FIG. 4, the parsing process 110 includes loading (120) the C program source code into memory 18. For each line of C program code, process 110 determines (122) whether the parser is at the start of a comment line. In the C language, a comment begins with the special symbols "*". If no comment start is detected, the process 110 replaces (124) global symbol names with hyperlinks and may color keywords according to the syntax. If a comment start is detected, the process 110 determines (126) whether a comment end is detected. In the C language, a comment ends with the special symbols "*\". If no comment end is detected, the process 110 finds (128) a block path within the comment by applying a multiple pattern match. If a block path is detected, the process 110 replaces (130) the block path with a special hyperlink back to the model diagram. This special hyperlink contains a command that highlights the references block, like the following example:

```
<A href = "execute:highight_system model/Block_A">model/BlockA</A>
```

The above command may be executed if the user 34 selects the hyperlink, and the user 34 has a browser capable of executing the command in the model diagram environment.

Referring to FIG. 5, an exemplary model diagram 190 and associated syntax highlighted code generation report 192 are

